



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

SOFTWARE PRO NÁVRH KONFIGURACE KAMER PANORAMATICKÉHO SYSTÉMU

SOFTWARE FOR CAMERAS CONFIGURATION DESIGN OF THE PANORAMIC SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Michal Jílek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Kříž

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

Student: Michal Jílek

Ročník: 3

ID: 174208

Akademický rok: 2017/18

NÁZEV TÉMATU:

Software pro návrh konfigurace kamer panoramatického systému

POKYNY PRO VYPRACOVÁNÍ:

Vytvořte aplikaci, která na základě vstupních požadavků uživatele (prostorové rozlišení výstupního obrazu, umístění kamer, ohniskové vzdálenosti objektivů apod.) pomůže určit klíčové parametry a konfiguraci objektivů a kamer, které jsou součástí panoramatického systému pro dohled nad sportovním stadionem ve vysokém rozlišení. Pomocí OpenGL vytvořte zjednodušený model tribun sportovního stadionu, jehož rozměry bude možno volit. Aplikace bude dále umožňovat např. generování virtuálního pohledu s vyznačením zorného pole z jednotlivých kamer, nastavení procentuálního překrytí jednotlivých záběrů tribun apod. Doporučeným vývojovým nástrojem je MS Visual C++ a balíček knihoven OpenGL.

DOPORUČENÁ LITERATURA:

[1] GUHA, S. Computer Graphics Through OpenGL: From Theory to Experiments. Chapman and Hall/CRC, 2010. ISBN 9781439846209.

[2] JOHNSON, B. Professional visual studio 2013: selected readings. 1st edition. Washington: Microsoft Press. Penguin education. ISBN 11-188-3204-3.

Termín zadání: 5.2.2018

Termín odevzdání: 29.5.2018

Vedoucí práce: Ing. Petr Kříž

Konzultant:

prof. Ing. Jiří Mišurec, CSc.

předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Software pro návrh kamer panoramatického systému“ jsem vypracoval

(-a) samostatně pod vedením vedoucího semestrální práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené semestrální práce dále prohlašuji, že v souvislosti s vytvořením této semestrální práce jsem neporušil(-a) autorská práva třetích osob, zejména jsem nezasáhl(-a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom (-a) následku porušení ustanovení § 11 a následujících autorského zákona c. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 29. 5. 2018

.....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Petru Křížovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

V Brně dne

Abstrakt

Základní fyzikální zákony o světle, princip snímání obrazu a parametry objektivů a kamer a vzorce pro výpočty jejich vhodného rozvržení na stadionu tak, aby snímaly dohromady panoramatický obraz. Ohnisková vzdálenost, zorný úhel, výpočty pokrytí, obrazové rozlišení. Přechod z teorie do praxe – návrh programu realizujícího danou problematiku. Základy jazyka C++ OpenGL. Návod na použití programu a ukázky.

Klíčová slova

C++ Open GL, kamery, obraz, panorama, průmyslové objektivy, rozlišení, stadion.

Abstract

The basic laws of physics about light, the principle of image capture, camera parameters and formulas for calculations of their appropriate layout of the stadium so that they read together a panoramic image. Focal length, viewing angle, coverage calculations, image resolution. The transition from theory to practice – draft programme of the issue. The fundamentals of the C++ language OpenGL. Program instructions and samples.

Keywords

C++ Open GL, camera, industrial lenses, panorama, picture, resolution, Stadium.

Obsah

| | | |
|-----|--|----|
| 1 | Základy optiky..... | 8 |
| 1.1 | Základy optiky..... | 8 |
| 1.2 | Čočky a promítání obrazu..... | 9 |
| 1.3 | Objektivy..... | 12 |
| 1.4 | Kamery a objektivy | 13 |
| 1.5 | Snímače | 14 |
| 2 | Návrh rozmístění kamer na stadionu | 16 |
| 2.1 | Přepočet ohniskové vzdálenosti na zorný úhel | 16 |
| 2.2 | Výpočet pokrytí obrazu kamerou | 17 |
| 2.3 | Stadion..... | 17 |
| 3 | Vytvoření panorama..... | 19 |
| 3.1 | Skládání panoramatických snímků | 19 |
| 4 | Návrh a provedení programu | 20 |
| 4.1 | Programovací jazyk C++ a specifikace OpenGL | 20 |
| 4.2 | Zprovoznění grafiky | 21 |
| 4.3 | Návrh programu | 21 |
| 5 | Závěr | 32 |

Úvod

V této semestrální práci na téma „Software pro návrh kamer panoramatického systému“ budou rozebrány hned 3 tematické okruhy, které je nezbytné pochopit, aby bylo možné vytvořit jeden celek. Celek bude program napsaný v programovacím jazyce C++, jehož úkolem bude přijímat vstupní data od uživatele a výstupem pak budou vypočtené parametry popisující vhodné rozložení kamer ve 2D. Bude nutné se seznámit se základními fyzikálními principy snímacích objektivů, rovnicemi popisujícími světlo a úhel záběru a z nich odvodit vztahy vyhovující danému objektivu.

Součástí této práce bude praktická část – tedy program provádějící výše zmíněné.

Použití tohoto programu najde uplatnění zejména pro návrhy rozmístění kamer na stadionech tak, aby byl zabrán stadion 360 ° kolem dokola.

1 Základy optiky

V této velké kapitole věnované danému tématu se bude pojednávat od úplných základů až

po samotné technické řešení daného problému. Teorie se skládá ze základních fyzikálních poznatků o světle, principem snímání a zobrazení obrazu, jeho digitalizace a komprese, dále pak o vybraných kamerách, jejich technických údajích, s rovnicemi a vzorci řešícími daný problém a se samotných programovacím jazyce C++ a o platformě Open GL. Po tomto rozboru se navrhne software řešící danou problematiku a popíše se jeho finální verze a manipulace s ním.

1.1 Základy optiky

1.1.1 Teorie světla

Světlo je příčné monochromatické charakteristické vlnovou délkou λ a frekvencí f . Bílé světlo je složené ze spektra barevných složek od červené po fialovou. Nachází se ve viditelné oblasti elektromagnetické spektra v pásmu 380 – 780 nm. Každé barvě přísluší určitá vlnová délka a frekvence. Pod hranicí tohoto pásu se nachází pásmo ultrafialového záření a nad hranicí pak infračervené záření. Základní vztah mezi vlnovou délkou, rychlostí a frekvencí udává vztah:

$$\lambda = v \cdot T = \frac{v}{f} [\text{Hz}]. \quad (1.1)$$

Barevný senzor je charakterizován spektrální citlivostí $s(\lambda)$. Barevné světlo je charakteristické pomocí spektrální funkce $r(\lambda)$. Odezva senzoru se pak vypočítá jako integrál:

$$o = \int_{\lambda_1}^{\lambda_2} s(\lambda)r(\lambda)d\lambda [-]. \quad (1.2)$$

Rychlost světla je konstanta ($c = 300\,000\text{ km/s}$). Ovšem ta se dá odvodit pomocí dvou konstant: permitivity ε a permeability μ :

$$c = \frac{1}{\sqrt{\varepsilon\mu}} = \frac{1}{\sqrt{\varepsilon_0\varepsilon_r\mu_0\mu_r}} \text{ [m/s]}, \quad (1.3)$$

kde ε_0 je konst., ε_r pak permitivita prostředí, μ_0 konst. a μ_r magnetická permeabilita.

Speciálně pro vakuum pak:

$$c = \frac{1}{\sqrt{\varepsilon_0\mu_0}} = 299\,792\,458 \text{ km/s}. \quad (1.4)$$

1.1.2 Barevný model RGB

Bílé světlo je tvořeno třemi složkami: R (red) – červená, G (green) – zelená a B (blue) – modrá. Vlnové délky těchto barev jsou: $\lambda_R = 630\text{ nm}$, $\lambda_G = 530\text{ nm}$, $\lambda_B = 450\text{ nm}$. Obecně platí, že

$$R + G + B = 1, \quad (1.5)$$

což znamená, že k určení třetí barvy stačí znát pouze dvě, přičemž třetí se dopočítá.

Zelená barva vykazuje nejvyšší jas, tudíž se v televizním vysílání nepřenáší, neboť je energeticky náročná, a lze ji dopočítat.

1.2 Čočky a promítání obrazu

1.2.1 Čočky a rozptylky

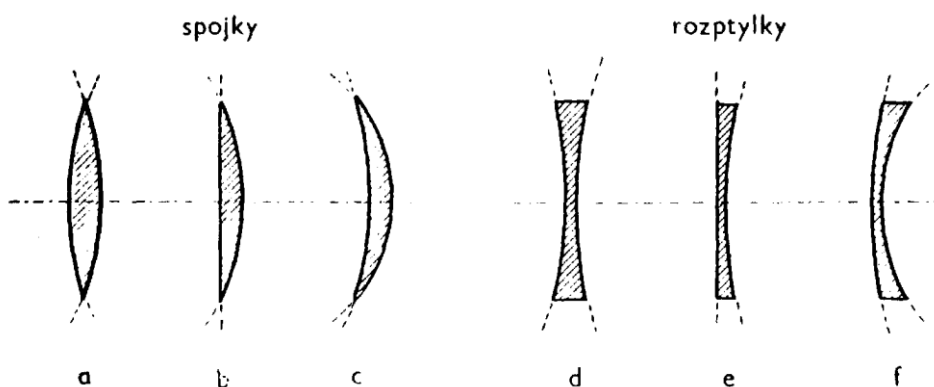
Optické čočky jsou součástí objektivů fotoaparátů a kamer. Jejich výroba je poměrně drahá. Důležitou roli hraje čistota a tvar čočky.

Platí zde známý Snellův zákon:

$$n = \frac{n_1}{n_2} = \frac{\sin \alpha}{\sin \beta}, \quad (1.6)$$

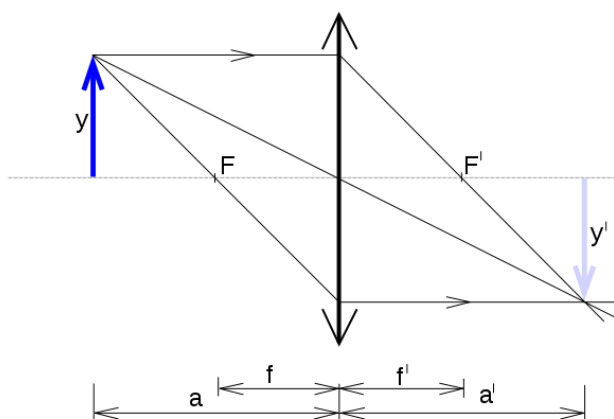
kde n představuje index lomu dvou prostředí a α a β pak úhel odrazu a lomu.

Rozdíl mezi čočkami a rozptylkami je takový, že čočky jsou vypuklé směrem ven. Obr. 1 ukazuje čočky a rozptylky:



Obr. 1: čočky a rozptylky; zdroj: http://kvinta-html.wz.cz/fyzika/optika/zobrazovani_optickymi_soustavami/obrazky/14.gif

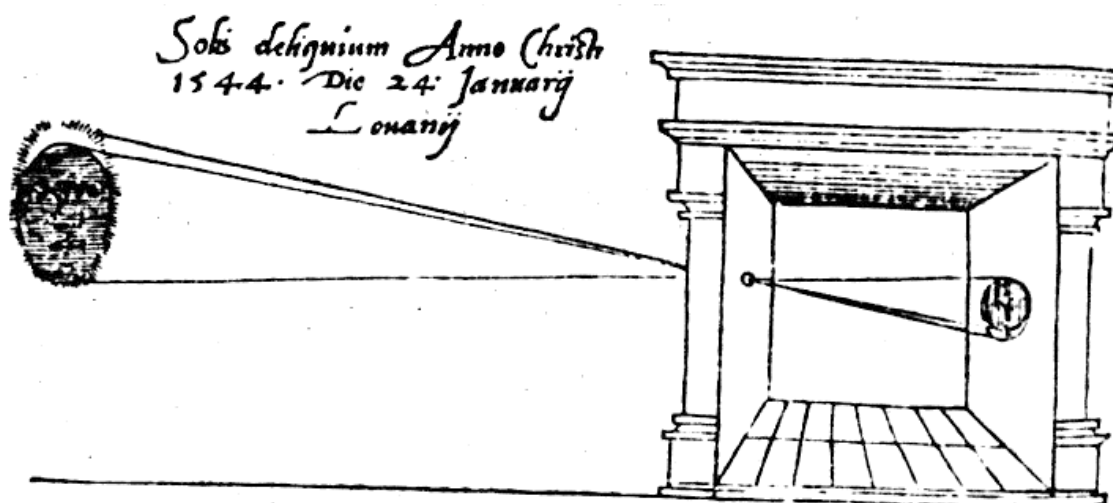
Obr. 2 pak ukazuje princip promítání obrazu čočkou:



Obr. 2: princip promítání čočkou

1.2.2 Objevení promítání

První objev dírkové komory se datuje 5. stol. př. n. l. Tohoto promítání se používá realistické zakreslování skutečnosti neboli obkreslování promítaného obrazu.



Camera obscura. R.G. Frisius, 1545

Obr. 3: nejstarší dírkové promítání

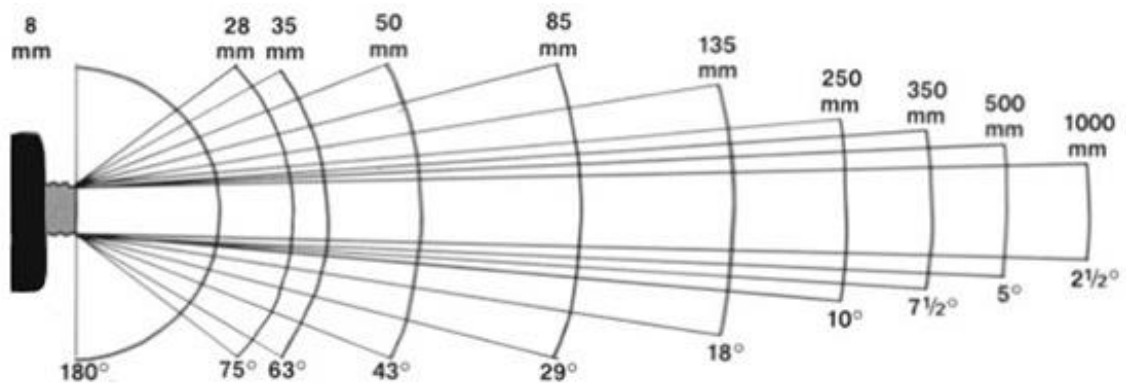
Jedná se v podstatě o jednoduché zobrazovací zařízení, v jehož stěně je malý otvor. Ten vytváří na protilehlé stěně obraz vnějšího prostoru na základě šíření přímočarého světla. Jakmile propuštěné paprsky narazí na průmětnu, vytvoří na ní obraz, avšak o 180° převrácený. Čím menší otvor, tím ostřejší je obraz. Jak vypočítat optimální průřez otvoru objevil Josef Petzval s tím, že výpočet zdokonalil Brit a držitel Nobelovy ceny Lord Rayleigh, který publikoval v roce 1891 v knize *Nature*. [1]

Nevýhodou je, že zachytává méně světla než čočky.

1.3 Objektivy

Ohnisková vzdálenost určuje úhel záběru – tedy oblast, kterou je objektiv schopný zachytit a přenést na snímač. Základní objektiv snímá po úhlem – 45° , což odpovídá vnímání lidského oka.

Objektivy s nejširším úhlem záběru – tedy 220° a 180° resp. 6 – 8 mm – dochází k výraznému zkreslení svislých a vodorovných linií. Jako širokoúhlé objektivy se označují ty, které mají ohniskovou vzdálenost 18 – 35 mm, kde se projevuje menší zkreslení na krajích obrazu. Základní objektivy mají nejmenší zkreslení. Ohniskové vzdálenosti – tzv. krátké teleobjektivy – se pohybují v rozsahu cca 85 – 350 mm. Objektivy pro speciální použití neboli dlouhé teleobjektivy mají ohniskové vzdálenosti cca 500 – 1000 mm – mají velmi malou hloubku ostrosti. [2]



Obr. 4: ohnisková vzdálenost / úhel záběru

1.4 Kamery a objektivy

Kamery USB3 Vision jsou jedny z mnoha, které lze použít pro náš účel. V této práci se zaměříme např. na model USB3 Vision kamera a mvBlueFox3 senzor.



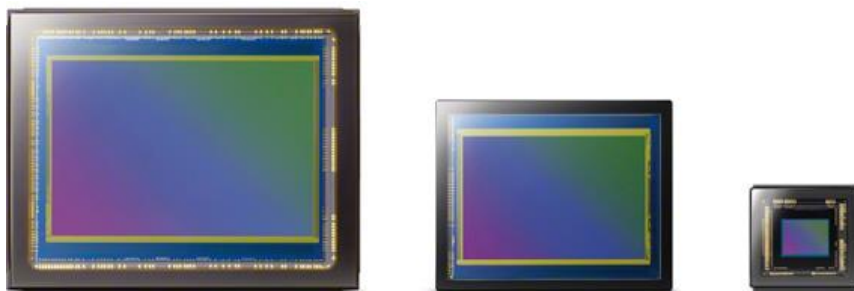
Obr. 5: USB3 Vision kamera a mvBlueFox3 senzor

Pro představu zde uvádím vybrané parametry tohoto modelu –1020 [3]:

- rozlišení: 1600×1200 (1,9 Mpx),
- maximální odstup signálu od šumu SNR_{MAX} : 38,8 dB,
- senzor: CMOS,
- připojení přes USB3: 5 Gb/s,
- optimální provozní teplota: $0 - 45\text{ }^{\circ}\text{C}$ (extrém: $-20 - 60\text{ }^{\circ}\text{C}$)

Velmi důležitý parametr každého objektivu je ohnisková vzdálenost. Ta se bude zadávat do vstupu konzole, která vypočítá, jak velkou plochu takový objektiv je schopen zabrat.

1.5 Snímače

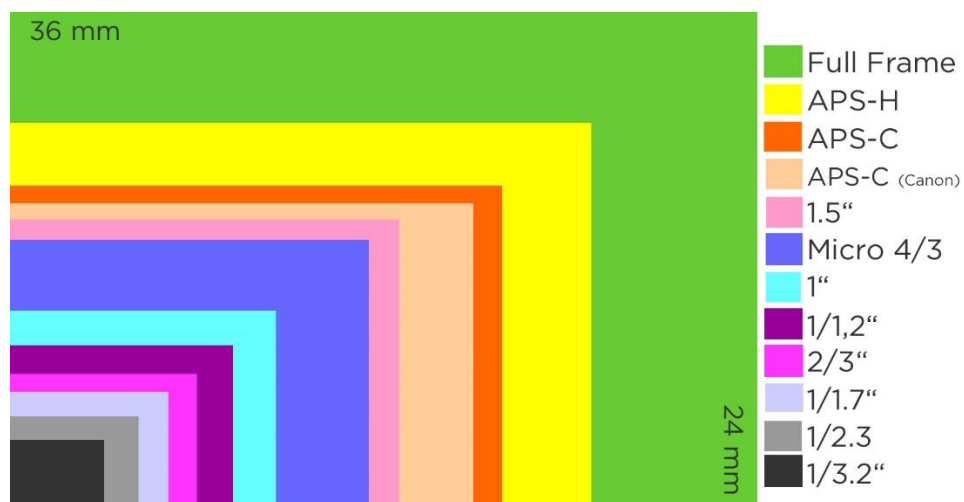


Obr. 6: snímače; https://www.fotolab.cz/blog/wp-content/uploads/2015/09/velikost_snimace_titulka_2.jpg

CCD

CMOS

1.5.1 Velikosti snímačů



Obr. 7: velikosti snímačů; https://www.fotolab.cz/blog/wp-content/uploads/2015/09/velikost_snimace.jpg

Tabulka 1: velikost a parametry snímačů

| Formát snímače | Rozměry [mm] | Crop faktor |
|----------------|-------------------|--------------|
| 1/3.2" | 4,54 x 3,42 | 7,6x |
| 1/3" | 4,80 x 3,60 | 7,2x |
| 1/2.3" | 6,17 x 4,55 | 5,6x |
| 1" | 13,20 x 8,80 | 2,7x |
| 4/3" | 17,30 x 13 | 2x |
| APS-C | 23,60 x 15,60 | 1,5x |
| APS-H | 27,90 x 18,60 | 1,3x |
| Full Frame | 36 x 24 | 1x |
| Středoformát | 33 x 44 mm a více | 0,79x a méně |

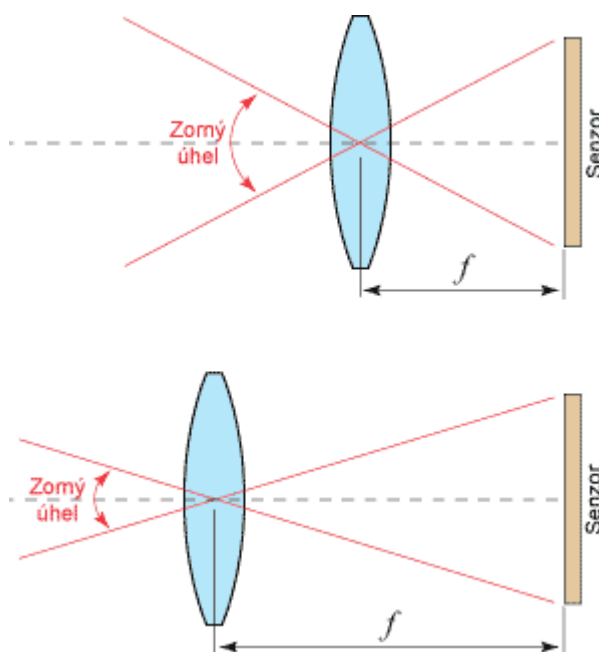
Crop faktor, neboli ekvivalentní ohnisková vzdálenost, umožňuje vypočítat ohniskovou vzdálenost jakou by měl objektiv se stejným zorným úhlem na Full Frame (35 mm) objektivu.

Zdroj: <https://www.fotolab.cz/blog/velikost-snimace/>

2 Návrh rozmístění kamer na stadionu

2.1 Přepočet ohniskové vzdálenosti na zorný úhel

Velmi důležitý vzorec pro přepočet ohniskové vzdálenosti daného objektivu na zorný úhel objektivu, který je součástí programu řešícího zadanou problematiku.



Obr. 8: Geometrické znázornění objektivu; <https://www.digimanie.cz/jak-je-to-s-crop-faktorem-objektivu-1dil/2252>

Z Obr. 8 je patrná následující rovnice:

$$\text{zorný úhel} = 2 \cdot \arctg \left(\frac{\text{rozměr senzoru}}{2 \cdot \text{ohnisková vzdálenost}} \right) [\text{mm}, \text{mm}, ^\circ] \quad (2.1)$$

K určení zorného úhlu, který je důležitý pro praktickou část, je potřeba znát samotný rozměr senzoru a udávanou ohniskovou vzdálenost, jak ukazuje rovnice (2.1).

2.2 Výpočet pokrytí obrazu kamerou

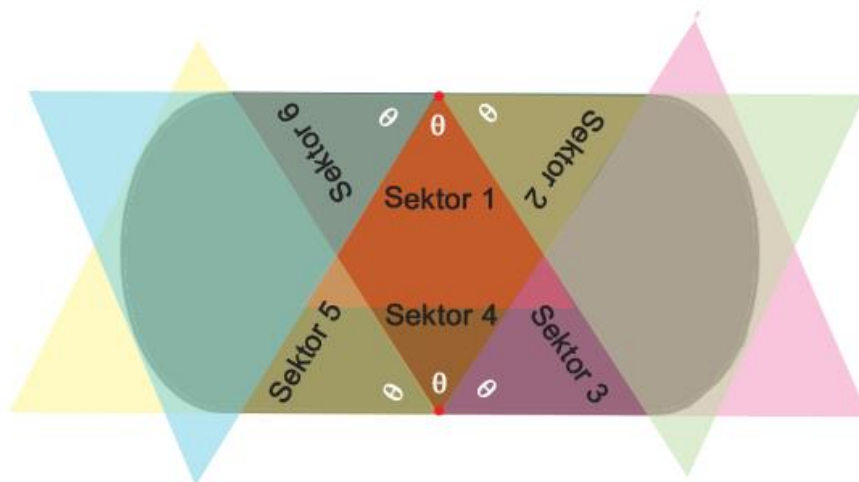
K tomu, abychom zjistili, kolik metrů pokryje konkrétní objektiv, je potřeba znát již známé parametry jako rozměr senzoru a ohnisková vzdálenost, ale také vzdálenost objektivu od protějšího konce stadionu, kam objektiv směřuje.

Vztah pro výpočet pokrytí ukazuje rovnice (2.2):

$$\frac{\text{šířka záběru}}{\text{vzdálenost záběru}} = \frac{\text{rozměr senzoru}}{\text{ohnisková vzdálenost}} [m, m, mm, mm] \quad (2.2)$$

2.3 Stadion

Uvažujme sportovní stadion, který má vnitřní obvod O a 2 body (červený puntík na Obr. 9: návrh rozmístění kamer na stadionu), kde budou nainstalovány kamery vedle sebe podél fiktivní půlkružnice. Každá kamera snímá pod úhlem θ svůj sektor. Cílem je kamery rozmístit tak, aby dohromady tvořily jeden obraz – panorama.



Obr. 9: návrh rozmístění kamer na stadionu

Počet kamer N , a tudíž i počet sektorů N závisí na mnoha faktorech. Tedy na snímacím úhlu kamery θ , vzdálenosti objektivu od obvodu, kam směřuje, a zároveň na vzdálenosti kamery od středu stadionu a na velikosti zoomu konkrétní kamery. Musí platit, že $N \times \theta = 360^\circ$. Již v tuto chvíli je zřejmé, že vytvořit přesné přechody mezi jednotlivými snímky nebude vůbec jednoduché a bude se to odvíjet od zcela přesných výpočtů a pečlivosti montáže a nastavení. O vytvoření panorama se už stará specifický software, který transformuje souřadnice rovin. Více panorama dále v této práci.

3 Vytvoření panorama

Panorama je záležitostí transformací několika obrazových rovin do jedné, aby vznikl jeden široký snímek. Jedná se o složité transformační matice, které přepočítávají sousedící snímky a snaží se je „přilepit“ vedle sebe. Je to analogie, jako kdybychom pořídili několik fotografií a slepili je k sobě, abychom získali širší záběr nebo panorama 360 °.

3.1 Skládání panoramatických snímků

3.1.1 Homografie

Definována jako perspektivní transformace mezi dvěma rovinami. Transformuje souřadnice (X, Y, Z) jedné roviny do souřadnic (X, Y, Z) druhé roviny.

3.1.2 Algoritmy

Existuje několik algoritmů skládání panoramatických snímků [4]:

- skládání pomocí homografie (vhodné pro rotující kamery),
- skládání pomocí válcových souřadnic a
- skládání pomocí kulových souřadnic.

4 Návrh a provedení programu

4.1 Programovací jazyk C++ a specifikace OpenGL

V této části se seznámíme s programováním grafických aplikací obecně. Potom se zaměříme na konkrétní program, který bude plnit účel zadání této práce.

Vývoj programu bude v prostředí MS Visual Studio 2018.

4.1.1 Jazyk C++

Programovací jazyk C++ je objektové rozšíření jazyka C. Jinými slovy, C++ je objektově orientovaný jazyk, na rozdíl od C. Tzn. zapouzdření, dědičnost a mnohotvárnost. V 90. letech 20. stol. došlo ke standardizaci jazyka normou ISO.

Výhodou jazyků C a C++ je rozšířenost a dostupnost pro všechny operační systémy (také PDA, mobilní telefony), či dokonce tam, kde se o operačním systému hovořit nedá (jednočipové mikropočítače). Jedná se o nízkoúrovňový jazyk, který má blízko ke strojovému kódu.

Moderními rysy jazyka C++ je např. ošetření výjimek a zavedení prostoru jmen. [5]

4.1.2 OpenGL

Jedná se o specifikaci definovanou API (Application Programming Interface) pro tvorbu 2D a 3D grafiky nezávisle na operačním systému, hardwaru a programovacím jazyku.

GLU (OpenGL Utility Library) je knihovna funkcí, které by měly být součástí instalací vývojových prostředí.

GLUT (OpenGL Utility Toolkit) je rovněž knihovna, které je díky své jednoduchosti vhodná pro začátečníky. Poskytuje jednoduché příkazy, jež zprostředkovávají funkce závislé na operačním systému (otevírání oken, interakce s uživatelem, ...). Tato knihovna není součástí vývojových prostředí a musí se nainstalovat. [6]

V této práci se budeme zabývat pouze vykreslováním do 2D roviny, kde zobrazíme jednoduché scény.

4.2 Zprovoznění grafiky

Ať už je program napsaný jako konzolová aplikace, nebo jako aplikace GUI, vykreslení grafiky funguje na principu vytvoření nového okna, kde se vykreslí naše příkazy.

Aby se grafické okno vytvořilo, je důležité zařadit do hlavní funkce programu main() následující kód:

```
47 int main(int argc , char * argv [])
48 {
49     |
50     glutInit(&argc, argv ); // inicializace knihovny GLUT
51
52     glutInitWindowSize(500, 400); // pocatecni velikost
53     glutInitWindowPosition(200, 200); // pocatecni pozice
54
55     window1 = glutCreateWindow("Graficke okno"); // vytvoreni okna
56     glutSetWindow(window1); // nastavit jako aktualni
57
58     glutDisplayFunc(onDisplay);
59     glutReshapeFunc(onResize);
60
61     glutMainLoop(); // nekonecna smycka
62
63
64
65     return 0;
66 }
```

Obr. 10: kód ve těle funkce main() pro zprovoznění vykreslování grafiky

4.3 Návrh programu

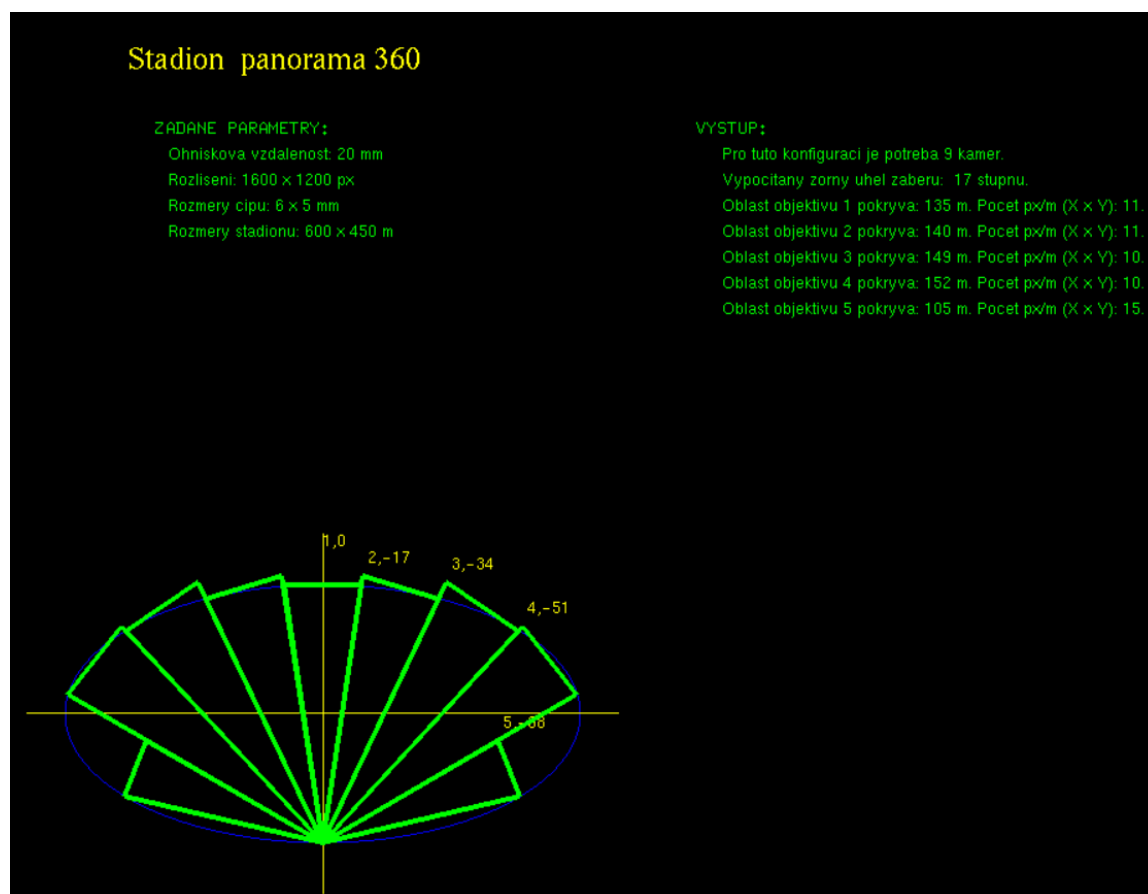
4.3.1 Popis programu – úvod

Jedná se o konzolovou aplikaci, která slouží jako software pro problematiku snímání panorama sportovních stadionů. Program bude navádět uživatele krok po kroku.

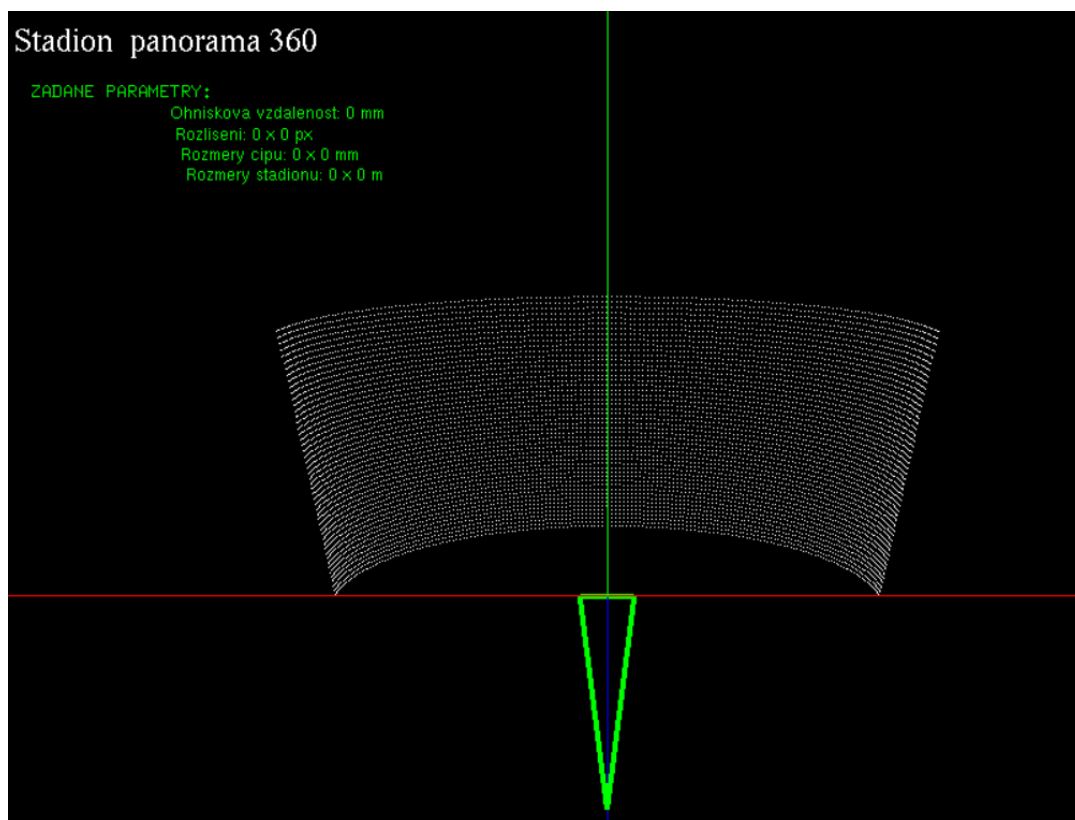
Jako uživatelský vstup program přejímá délku a šířku stadionu a ohniskovou vzdálenost objektivu.

Data se přepočítají. Z ohniskové vzdálenosti se vypočítá úhel záběru, který bude využit pro výpočet rozměrů stran trojúhelníků. Naleznou se průsečíky přímky s elipsou, což bude charakterizovat úhel záběru objektivu a fyzický stadion.

Výstupem programu bude pak vypočtený počet objektivů nutných pro zachycení panorama a grafické vykreslení jednotlivých záběrů objektivů, které budou pro přehlednost snímat pouze horní polovinu stadionu – tedy 180° .



Obr. 11: ukázka programu ve 2D



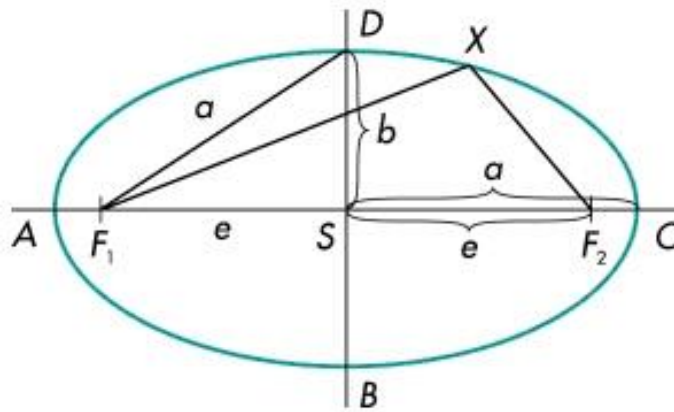
Obr. 12: Ukázka programu ve 3D

Jednotlivými matematickými operacemi, nutnými k dosažení požadovaného výsledku, se zabývají následující kapitoly.

4.3.2 Elipsa

Elipsa je kuželosečka. Její definice zní: „Elipsa je množina bodů, které mají konstantní součet vzdáleností od dvou daných pevných bodů“. Tuto definici popisuje rovnice:

$$|F1| + |F2| = 2a \quad (4.1)$$



Obr. 13: elpisa

Strana a se nazývá hlavní poloosa, b vedlejší poloosa, e excentricita neboli výstřednost. Platí:

$$a^2 = b^2 + e^2 \quad (4.2)$$

4.3.3 Průnik přímky s elipsou

Abychom mohli spočítat průnik přímky s elipsou, je potřeba znát rovnici přímky a elipsy. Jelikož pracujeme ve 2D, zajímají nás pouze souřadnice X a Y . Průsečíky budou 2, jak je vidět na obrázku. Vyřešení soustavy rovnice přímky a elipsy se najdou průsečíky v bodech x_1, y_1 a x_2, y_2 . Řešením musí být kvadratická rovnice se dvěma kořeny.

Obecná rovnice přímky:

$$ax^2 + b = y \quad (4.3)$$

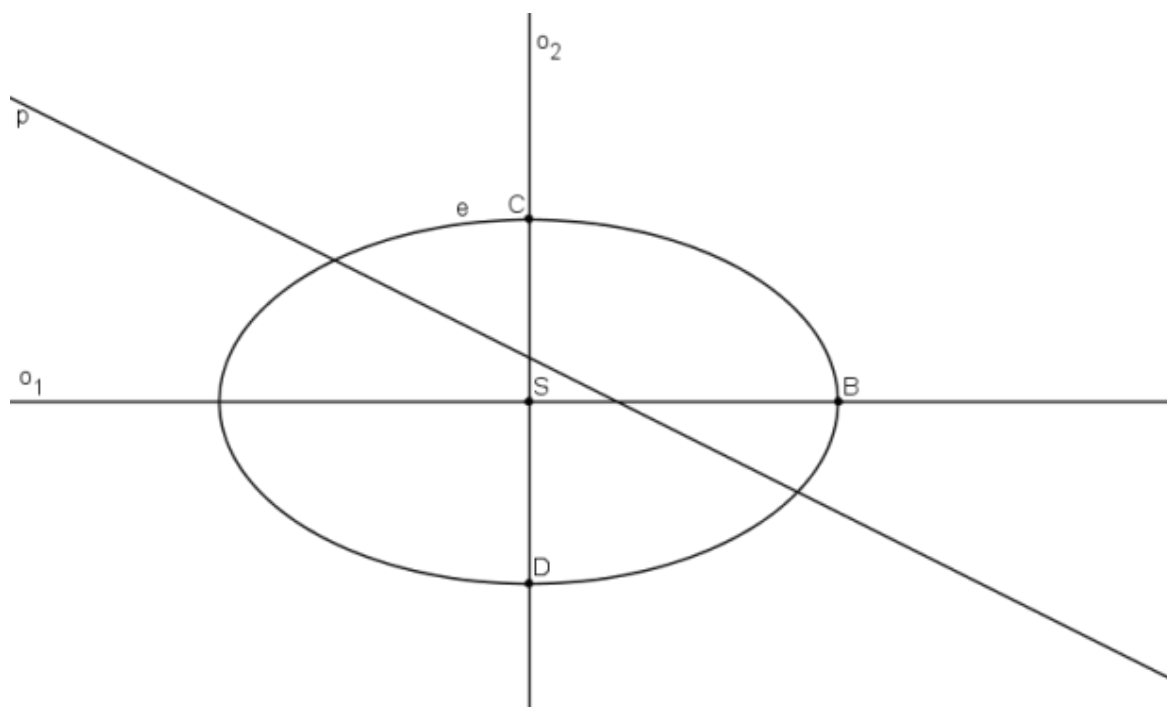
Obecná rovnice elipsy:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (4.4)$$

Pakliže elipsa nemá střed v počátku soustavy souřadné, tj. $S \neq [0,0]$, platí rovnice:

$$\frac{(x - m)^2}{a^2} + \frac{(y - n)^2}{b^2} = 1, \quad (4.5)$$

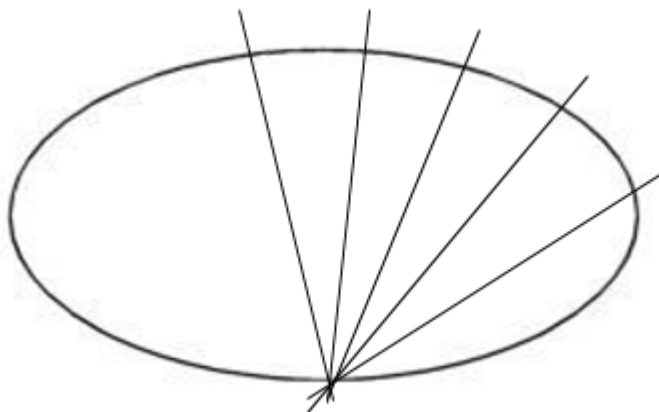
kde bod m a n jsou souřadnice středu elipsy. Tato rovnice tedy popisuje posunutí elipsy od počátku.



Obr. 14: Průnik přímky s elipsou obecně; zdroj: <http://www.karlin.mff.cuni.cz/~plichtova/Diplomka/AfinitaAKolineace/images/Pruseciky1.png>

Řešení v programu na míru:

Cílem zadání této práce je naprogramovat software, který na základě ohniskové vzdálenosti objektivu vypočte počet nutných objektivů pro zabránění panorama 360°. Kromě toho také vykreslí stadion včetně úhlu záběru až na úplný konec stadionu – neboli snímanou plochu. K tomu se využije platformy OpenGL, která vykresluje počítačovou grafiku. Jednotlivé body o souřadnicích x_i a y_i se nazývají vertexy. Podle toho, co chceme vykreslovat, specifikujeme tzv. mód primitiva. Pro propojení jednotlivých abstraktních vertexních bodů využívám primitiva `GL_LINES`. To se postará o vykreslení 3 čar a vznik trojúhelníku, který reprezentuje snímanou plochu daným objektivem.



Obr. 15: Přímka procházející spodní polovinou elipsy

Elipsa, která reprezentuje půdorys stadionu, má střed v bodech o souřadnicích $[m, n] = [250, 150]$. Proto pro výpočet její rovnice využijeme rovnice 4.5. Po dosazení za m, n, a a b dostaneme:

$$\frac{(x - 250)^2}{200^2} + \frac{(y - 150)^2}{100^2} = 1 \quad (4.6)$$

Společný bod všech přímek je bodem místa umístění kamer, které budou snímat protější stranu. V tomto bodě bude umístěno tolik kamer, aby dohromady snímali I. a II. kvadrant elipsy – tedy 180°. Podle úhlu záběru vznikne trojúhelník. Tento bod má abstraktní souřadnice $[x_0, y_0] = [250, 50]$, které jsou společné pro všechny přímky. Pro určení zobecněné rovnice přímky se 2 proměnnými je potřeba dosadit souřadnice x_1, y_1, x_2 a y_2 do rovnice 4.3. Potom se vypočítají koeficienty a a b , z nichž se odvodí následující rovnice:

$$y = \frac{50 - y_1}{250 - x_1} (x - 250) + 50 \quad (4.7)$$

Zde x_1 a y_1 představují čísla neboli proměnné, za které dosadí program souřadnice dané přímky. Jak je zjistí si povíme později.

Nyní budeme hledat průsečík přímky s elipsou. Dosadíme rovnici 4.7 do rovnice 4.6 a rovnici upravíme:

$$\begin{aligned} s &= (x^2 - 500x + 250^2) \cdot 100^2 \\ t &= 100^2 \cdot 200^2 \\ s + \left(\frac{50 - y_1}{250 - x_1} \cdot (x - 250) - 100 \right)^2 \cdot 200^2 &= t \\ s + 200^2 \left(\frac{(50 - y_1)^2}{(250 - x_1)^2} \cdot (x - 250)^2 - 200 \frac{(50 - y_1)(x - 250)}{250 - x_1} + 100^2 \right) &= t \\ s + 200^2 \frac{(50 - y_1)^2 \cdot (x^2 - 500x + 250^2)}{(250 - x_1)^2} - 200^3 \frac{50x - 50 \cdot 250 - xy_1 + 150y_1}{250 - x_1} &= t \\ + 100^2 200^2 &= 100^2 \cdot 200^2 \end{aligned}$$

Vzhledem k složitým úpravám využívám často substituce:

$$e = 100^2 (250 - x_1)^2 \quad (4.8)$$

$$f = 200^2 (50 - y_1)^2 \quad (4.9)$$

$$g = 200^3 (250 - x_1) \quad (4.10)$$

Další mezikroky:

$$\begin{aligned} x^2 e - 500 x e + 250^2 e + x^2 f - 500 x f + 250^2 f - 50 x g + 50 \cdot 250 g \\ xy_1 g - 250 y_1 g + 100^2 200^2 (250 - x_1)^2 = 100^2 200^2 (250 - x_1)^2 \end{aligned}$$

$$x^2(e + f) + x(-500e - 500f - 50g + y_1g) + 250^2e + 250^2f + 50.250g - 250y_1g = 0$$

Jak je vidět, vznikla očekávaná kvadratická rovnice $ax^2 + bx + c = 0$. Příslušné koeficienty uložíme do proměnných a , b a c :

$$a = e + f \quad (4.11)$$

$$b = -500e - 500f - 50g + y_1g \quad (4.12)$$

$$c = 250^2e + 250^2f + 50.250g - 250y_1g \quad (4.13)$$

Nyní je potřeba spočítat kořeny kvadratické rovnice:

$$X_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (4.14)$$

$$X_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (4.15)$$

Ty se pak dosadí do některého z mezikroků a získáme ypsilonové souřadnice:

$$Y_1 = \frac{50 - y_1}{250 - x_1}(X_1 - 250) + 50 \quad (4.16)$$

$$Y_2 = \frac{50 - y_1}{250 - x_1}(X_2 - 250) + 50 \quad (4.17)$$

Nalezené kořeny jsou klíčovou rovnicí, která řeší naši problematiku. Rovnice 4.8 – 4.17 jsou napsány v programu:

```

4      e = 100*100*(250-x1)*(250-x1);
5      f = 200*200*(50-y1)*(50-y1);
6      g = 200*200*200*(250-x1);
7
8      a = e+f;
9      b = -500*e-500*f-50*g+y1*g;
0      c = 250*250*e+250*250*f+50*250*g-250*y1*g;
1
2      xx1 = (-b+sqrt(b*b-4*a*c))/(2*a);
3      xx2 = (-b-sqrt(b*b-4*a*c))/(2*a);
4
5      yy1 = (50-y1)*(xx1-250)/(250-x1) + 50;
6      yy2 = (50-y1)*(xx2-250)/(250-x1) + 50;
7

```

Obr. 16: ukázka praktického řešení hledání průsečíku v programu

Vždy jeden kořen X_i a Y_i bude procházet stanoveným bodem [250,150]. Tento kořen nás ale nezajímá. Zajímá nás ten druhý kořen X_i a Y_i , což je bod, kde se strana trojúhelníku protne s elipsou. Ty nezajímavé kořeny se musejí vyloučit podmínkou.

Speciální případ tvoří vykreslení prvního trojúhelníku. Jedná se totiž o rovnoramenný trojúhelník, jehož výška je 2násobek vedlejší poloosy, tedy $2b$.

4.3.4 Ukládání souřadnic do polí

Princip založen na ukládání vypočtených os X a Y do dvourozměrného pole typu double. Rozměr pole se vypočítá na základě údaje o úhlu, který svírá bod umístění objektivů s bodem průniku hlavní poloosy a elipsy. Jednotlivé objektivy se mohou posouvat jak v horizontálním, tak i ve vertikálním směru. Vzorec pro výpočet minimálního úhlu ke zjištění rozměru pole: ...

V případě prvního trojúhelníku reprezentujícího první objektiv stačí znát pouze dvoje souřadnice X , Y . Proto bude toto pole mít dva řádky a dva sloupce. Ostatní trojúhelníky reprezentující další objektivy vyžadují čtyři souřadnice X , Y , neboť se jedná o dva souměrné trojúhelníky podle vedlejší poloosy. (Dva souměrné trojúhelníky proto, neboť jsou zadány jako požadovány vedoucím práce.) Proto bylo zvoleno pole o čtyřech řádcích a dvou sloupcích.

Příklad vytvořeného pole pro objektivy s ohniskovou vzdáleností 10 mm, s 5mm čipem a s umístěním na středu stadionu na jeho nejvzdálenějším bodě vypadá následovně:

| | 0 | 1 | |
|----|--------|--------|----------------|
| 0 | 281,25 | 250,00 | 1. trojúhelník |
| 1 | 218,75 | 250,00 | |
| 2 | 282,50 | 257,99 | 2. trojúhelník |
| 3 | 344,40 | 238,16 | |
| 4 | 217,50 | 257,99 | |
| 5 | 155,60 | 238,16 | |
| 6 | 351,25 | 251,82 | 3. trojúhelník |
| 7 | 407,99 | 211,32 | |
| 8 | 148,75 | 251,82 | |
| 9 | 92,01 | 211,32 | |
| 10 | 408,02 | 211,34 | 4. trojúhelník |
| 11 | 449,70 | 155,50 | |
| 12 | 91,98 | 211,34 | |
| 13 | 50,30 | 155,50 | |

Princip je z obrázku patrný a vypadá jednoduše. Nicméně zrealizovat jej bylo velmi náročné, neboť je důležité správné uložení souřadnic v celém poli, aby mohli být jednotlivé vertexy pospojovány správně do trojúhelníku. Při jakékoliv chybě se nebudou vertexy vykreslovat správně a bude to celé „rozhozené“.

Celkově je zde složitá práce s iterátory uvnitř cyklu.

Platí: čím více mm ohnisková vzdálenost, tím vznikne více trojúhelníků a s nimi i více řádků v poli.

Algoritmus

Uživatel zadá délku a šířku stadionu a ohniskovou vzdálenost objektivu.

Program přepočte ohniskovou vzdálenost úhel záběru.

Z úhlu záběru a rozměrů stadionu se vypočítá plocha záběru.

Plochy záběru se počítají tak dlouho, dokud nepokryjí alespoň polovinu obvodu stadionu – tedy 180° , čímž se zjistí počet nutných pro zabrání poloviny panorama.

Počet objektivů se vynásobí dvěma.

Uživatel obdrží výstup – tedy počet objektivů pro snímání panorama.

Do nového okna se vykreslí stadion a kamery zabírající horní polovinu stadionu.

5 Závěr

Procvičil jsem si zde pořádně analytickou geometrii, prohloubil jsem znalosti programovacího jazyka C++, dozvěděl jsem se o počítačové grafice v OpenGL, parametry objektivů a v neposlední řadě principy promítání. Navíc jsem zde popsal své základní znalosti z fyziky o optice, která zde hraje velkou roli.

Tato práce – zejména praktická část je plná matematiky a jsem rád, že mi to nedělalo problémy. Jen trvalo delší dobu, než jsem vyřešil dlouhou soustavu rovnic přímky a elipsy, ale nakonec se podařilo a s výsledky jsem spokojený.

Hlavním cílem bylo vytvořit program, který vypočítá počet objektivů pro snímání panorama na základě jejich údajů a vykreslí to graficky.

Bohužel neměl jsem dostatek času na to, abych jak teoretickou, tak i praktickou část dokončil na 100 %. Tvorba programu zabrala mnoho času, který nebyl z důvodu jiných školních povinností, mimoškolních povinností, a také z životního a osobního vytížení. I tak jsem věnoval práci hodně času a jsem rád, že i přes všechny tyto překážky se mi to celé podařilo dostat do téměř finální podoby. Na práci a samotném programu je stále co zlepšovat. Předem děkuji za pochopení a vžití se do nelehké situace.

A PŘÍLOHY

Konzolová aplikace „BP_Stadion_panorama_1.exe“.

Zdrojový kód projektu „BP_Stadion_panorama_1“.

Konzolová aplikace „BP_Stadion_panorama_2.exe“.

Zdrojový kód projektu „BP_Stadion_panorama_2“.

Seznam obrázků

| | |
|---|----|
| Obr. 1: čočky a rozptylky; zdroj: http://kvinta-html.wz.cz/fyzika/optika/zobrazovani_optickymi_soustavami/obrazky/14.gif | 10 |
| Obr. 2: princip promítání čočkou | 10 |
| Obr. 3: nejstarší dírkové promítání | 11 |
| Obr. 4: ohnisková vzdálenost / úhel záběru | 12 |
| Obr. 5: USB3 Vision kamera a mvBlueFox3 senzor | 13 |
| Obr. 6: snímáče; https://www.fotolab.cz/blog/wp-content/uploads/2015/09/velikost_snimace_titulka_2.jpg | 14 |
| Obr. 7: velikosti snímáčů; https://www.fotolab.cz/blog/wp-content/uploads/2015/09/velikost_snimace.jpg | 14 |
| Obr. 8: Geometrické znázornění objektivu; https://www.digimanie.cz/jak-je-to-s-crop-faktorem-objektivu-1dil/2252 | 16 |
| Obr. 9: návrh rozmístění kamer na stadionu | 17 |
| Obr. 10: kód ve těle funkce main() pro zprovoznění vykreslování grafiky | 21 |
| Obr. 11: ukázka programu ve 2D | 22 |
| Obr. 12: Ukázka programu ve 3D | 23 |
| Obr. 13: elpisa | 24 |
| Obr. 14: Průnik přímky s elipsou obecně; zdroj: http://www.karlin.mff.cuni.cz/~plichtova/Diplomka/AfinitaAKolineace/images/Pruseciky1.png | 25 |
| Obr. 15: Přímka procházející spodní polovinou elipsy | 26 |
| Obr. 16: ukázka praktického řešení hledání průsečíku v programu | 29 |

Reference

- [1] BALIHAR, David. CO JE DÍRKOVÁ KOMORA. *Pinhole.cz* [online]. , 1 [cit. 2017-10-14]. Dostupné z: <<http://www.pinhole.cz/cz/pinholecameras/whatis.html>>.
- [2] MACENAUER, Andrej. Základní charakteristiky objektivů I - Lek. 8. *Fotorádce* [online]. 2001, **2001**(8), 1 [cit. 2017-11-26]. Dostupné na: <<https://www.fotoaparát.cz/clanek/219/zakladni-charakteristiky-objektivu-i-lek-8-5007/>>.
- [3] *USB3 Vision camera with e2v / Aptina sensors - mvBlueFOX3* [online]. [cit. 2017-12-14]. Dostupné z: <<https://www.matrix-vision.com/USB3-vision-camera-mvbluefox3.html>>.
- [4] Skládání panoramatických snímků. ŘÍHA, Kamil, doc.Ing.,PhD. *Pokročilé techniky zpracování obrazu*. Brno: FEKT, VUT Brno, 2013, s. 143. ISBN 978-80-214-4894-0.
- [5] ŠALOUN, Petr, doc. RNDr.,PhD. *Programovací jazyk C++ pro zelenáče*. Jihlava: Neocortex, 2005. ISBN 80-86330-18-4.
- [6] RAJMÍČ, Pavel, doc.Mgr.,PhD. *Grafické a multimediální procesory – počítačová cvičení*. Brno: FEKT, VUT Brno, 2012. ISBN 978-80-214-4637-3.